View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ePrints.FRI

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Amon Stopinšek

Uporaba globokih nevronskih mrež za ločevanje avtomatsko generiranih in ročno napisanih člankov

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: prof. dr. Igor Kononenko

Ljubljana, 2019

Copyright. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Naloga je preizkusiti različne vrste globokih nevronskih mrež na problemu ločevanja avtomatsko zgeneriranih člankov od člankov, ki jih je napisal človek. Pri tem se omejite vsebinsko na eno zvrst člankov, npr. na članke o hujšanju. Podajte tudi kratek pregled obstoječih pristopov za klasifikacijo besedil. Preizkusite tudi različne predstavitve besedil za vhod v nevronske mreže. Ločite med znakovno in besedno predstavitvijo. Za besedne predstavitve preizkusite nekaj najpogosteje uporabljenih prednaučenih besednih vložitev.

Zahvaljujem se mentorju prof. dr. Igorju Kononenku za pomoč pri strukturiranju diplomske naloge in ostale strokovne nasvete. Zahvaljujem se tudi staršem, prijateljem in vsem ostalim, ki so tako ali drugače pripomogli k izdelavi diplomske naloge.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
BLSTM	bidirectional long short-term memory	dvosmerna LSTM nevronska mreža
BRNN	bidirectional recurrent neural network	dvosmerna rekurenčna nevronska mreža
CA	classification accuracy	klasifikacijska točnost
CBOW	continuous bag of words	neprekinjena vreča besed
CNN	convolutional neural network	konvolucijska nevronska mreža
C-LSTM	convolutional long short-term memory	konvolucijska LSTM nevronska mreža
DNN	deep neural network	globoka nevronska mreža
GRU	gated recurrent units	rekurenčni modul nevronske mreže z vrati
KNN	k-nearest neighbors	k-najbljižih sosedov
LSTM	long short-term memory	nevronske mreže z dolgim kratkoročnim spominom
NLTK	natural language toolkit	zbirka orodij za procesiranje naravnega jezika
RCNN	recurrent convolutional neural network	rekurenčna konvolucijska nevronska mreža
ReLU	rectified linear unit	pragovna linearna funkcija
RNN	recurrent neural network	rekurenčna nevronska mreža
SVM	support vector machine	metoda podpornih vektorjev

Povzetek

Naslov: Uporaba globokih nevronskih mrež za ločevanje avtomatsko generiranih in ročno napisanih člankov

Avtor: Amon Stopinšek

V diplomskem delu se ukvarjamo s klasifikacijo besedil na problemu ločevanja člankov, ki jih je napisal človek, od člankov, ki jih je napisal stroj. Na problemu smo preizkusili različne arhitekture konvolucijskih in rekurenčnih globokih nevronskih mrež ter različne predstavitve besedil. Modele smo testirali na podatkovni zbirki ročno napisanih in generiranih člankov o hujšanju. Najboljše rezultate smo dosegli z uporabo arhitekture BLSTM z besednimi vložitvami *word2vec*. S tem modelom smo dosegli $96,71\%$ klasifikacijsko točnost na testni podatkovni množici na ročno napisanih člankih, 100% klasifikacijsko točnost na člankih, generiranih s slabim modelom in $97,41\%$ klasifikacijsko točnost na člankih, generiranih z dobrim modelom.

Ključne besede: umetna inteligenca, globoke nevronske mreže, klasifikacija besedil, procesiranje naravnega jezika.

Abstract

Title: Using deep neural networks for differentiating automatically generated from manually written articles

Author: Amon Stopinšek

This thesis deals with the text classification on the problem of classifying manually written and automatically generated articles. We tested various convolutional and recurrent deep neural network architectures and various text representations. Models were tested on a dataset of manually written and automatically generated articles about weight loss. Best results were achieved with a model using the BLSTM architecture and *word2vec* word embeddings. With this model, we achieved $96,71\%$ classification accuracy on the test dataset of manually written articles, 100% classification accuracy on articles generated with the bad model and $97,41\%$ classification accuracy on articles generated with the good model.

Keywords: artifical inteligence, deep neural networks, text classification, natural language processing.

Poglavje 1

Uvod

Z napredkom nevronskih mrež [12] so se začeli pojavljati modeli za generiranje besedil [38]. Generirana besedila se ob branju hitro loči od človeško napisanih. Kljub slabi kvaliteti pa je predvsem na spletu precej možnosti za uporabo takšnih besedil. Generirana besedila bi se lahko naprimer uporabila za članke, namenjene optimizaciji spletnih strani, recenziji izdelkov in storitev [24] ali grajenju profilov na forumih in socialnih omrežjih [13].

Zaradi pomanjkanja nadzora nad verodostojnostjo generiranega besedila in preproste avtomatizacije generiranja in objave je pomembno, da imamo na voljo učinkovit model, ki takšna besedila zazna. Po drugi strani lahko model, ki dobro ločuje generirana besedila od ročno napisanih, uporabimo za izboljšavo modela za generiranje besedil.

Ločevanje avtomatsko generiranih in ročno napisanih člankov je problem klasifikacije besedil v dve kategoriji. Od najbolj tipičnih primerov klasifikacije besedil, kot je na primer zaznavanje neželene elektronske pošte, se loči v tem, da se pri klasifikaciji modeli ne morejo zanašati na frekvenco določenih besed ali besednih zvez.

Glavni cilj diplomske naloge je preizkus in primerjava različnih arhitektur globokih nevronskih mrež na problemu ločevanja ročno napisanih in generiranih člankov. V 2. poglavju smo predstavili problem klasifikacije besedil, opisali postopek grajenja modela za klasifikacijo in predstavili nekaj mode-

lov za klasifikacijo besedil z globokimi nevronskimi mrežami. V 3. poglavju so opisane uporabljene arhitekture nevronskih mrež in predstavitve besedil. V 4. poglavju smo predstavili uporabljeno podatkovno množico, zgrajene modele, čase učenja in klasifikacije modelov ter rezultate.

Poglavje 2

Klasifikacija besedil

Klasifikacija besedil je problem procesiranja naravnega jezika. V praksi ima problem širok nabor uporabe, kot je zaznavanje neželene elektronske pošte [21], zaznavanje sentimenta [30], ocenjevanje/rangiranje besedil [1], označevanje dokumentov [25].

Problem se najpogosteje rešuje z dvema različnima pristopoma - z uporabo klasičnih metod strojnega učenja (SVM, KNN, naivni Bayes) [14] ali z nevronskimi mrežami z uporabo različnih arhitektur (CNN, RNN) [40].

Grajenje modelov za klasifikacijo besedila sestoji iz večih korakov:

1. priprave učne in testne množice:

   Celotno množico podatkov razdelimo na učno in testno množico. Običajno 70% podatkov namenimo za učno množico, preostalih 30% pa za testno množico.

2. normalizacije besedila:

   Iz besedila odstranimo odvečne dele, poenotimo strukturo in dodamo dodatne oznake [41, 42]. Najpogostejše operacije v tem koraku so:

   - odstranitev označevalnih značk (HTML, XML),
   - pretvorba črk z veliko začetnico v malo,
   - poenotenje predstavitve numeričnih vrednosti,

- odstranitev neinformativnih besed (ang. *stopwords*),
- stavčna segmentacija,
- tokenizacija,
- lematizacija,
- dodajanje oblikoskladenjskih oznak.

3. predstavitev podatkov:

   V tem koraku besedilo pretvorimo v obliko, primerno za učenje modela. Odvisno od arhitekture mreže lahko besede le nadomestimo z indeksom iz slovarja vseh besed, besede predstavimo v obliki vloženih vektorjev ali celotno besedilo pretvorimo v vrečo besed.

4. učenja modela (glej razdelek 2.1),

5. evalvacije modela:

   Točnost modela je potrebno oceniti. V ta namen model uporabimo za napovedovanje razredov v testni množici.

   Klasifikacijska točnost (2.1) predstavlja delež pravilno klasificiranih primerov od vseh napovedi. Mera v primeru neuravnotežene razporeditve med razredi slabo prikaže natančnost modela. V dvorazrednih klasifikacijskih problemih se jo lahko izračuna kot:

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

   kjer TP predstavlja število pravilno klasificiranih pozitivnih primerov, TN število pravilno klasificiranih negativnih primerov, FP število napačno klasificiranih negativnih primerov v pozitivne in FN število napačno klasificiranih pozitivnih primerov v negativne.

   Matrika zmot v obliki tabele prikazuje število pravilnih in napačnih napovedi razporejenih po razredih. Uporabna je predvsem v primeru neuravnoteženih razredov. Tabela 2.1 prikazuje matriko zmot pri dvorazrednem problemu, kjer je s P označen pozitivni razred in z N negativni.

		napovedani razred	
		P	N
resnični razred	P	TP	FN
	N	FP	TN

Tabela 2.1: Matrika zmot pri dvorazrednem problemu, kjer je s P označen pozitivni razred in z N negativni.

2.1 Učenje modela

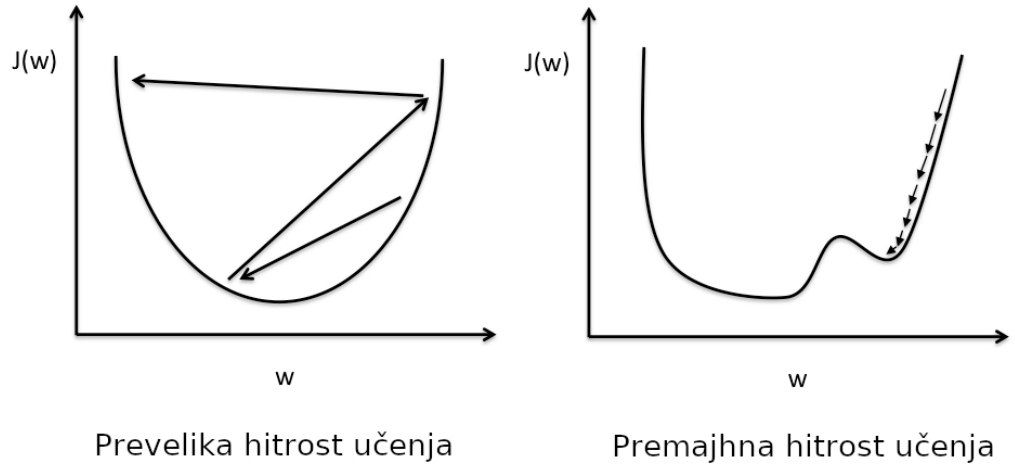
Pri učenju nevronskih mrež je zelo pomembna pravilna nastavitev hiper parametrov [2]. Hiper parametri določajo strukturo nevronske mreže (npr. število skritih plasti) in postopek učenja (npr. hitrost učenja).

2.1.1 Hitrost učenja

Hitrost učenja (ang. *learning rate*) določa, s kakšnim korakom popravimo uteži v smeri minimuma. V primeru premajhne hitrosti učenja bo učenje modela počasno, lahko se zgodi, da obtičimo v lokalnem minimumu. Če je hitrost učenja prevelika, pride to prevelikih sprememb uteži, globalnega minimuma pa zaradi tega ne dosežemo. Na sliki 2.1 sta prikazana primera prevelike in premajhne hitrosti učenja.

2.1.2 Velikost paketa

Velikost paketa (ang. *batch size*) določa, koliko učnih primerov bo šlo skozi mrežo v enem koraku vzvratnega razširjanja napake. Večja velikost paketa bolje oceni gradient, a porabi več pomnilnika [8] in lahko vodi v slabšo posplošitev modela [20].



Slika 2.1: Primera prevelike in premajhne učne hitrosti [33].

2.1.3 Število učnih iteracij

Število učnih iteracij (ang. *epoch*) predstavlja kolikokrat modelu pokažemo vse učne primere. V primeru premajhnega števila učnih iteracij dobimo slabši model, če pa jih je preveč, lahko pride do prevelikega prileganja učnim podatkom (ang. overfitting).

2.2 Klasifikacija besedil z DNN

Na področju klasifikacije besedil z globokimi nevronskimi mrežami (ang. *Deep Neural Networks* (DNN)) prevladujejo modeli, osnovani na konvolucijskih nevronskim mrežah (CNN) in rekurenčnih nevronskih mrežah (RNN) [40].

Besedila so predstavljena kot zaporedja besed, preslikanih v besedne vložitve (glej razdelek 3.4) ali kot zaporedje znakov predstavljenih v obliki vektorjev kodiranih z uporabo kodiranja $one - hot$. Kodiranje $one - hot$ kategorične podatke predstavi v obliki vektorjev, kjer ima en element vektorja vrednost 1, vsi ostali elementi pa vrednost 0. Število različnih vrednosti v kategoriji

določa dolžino vektorja [22].

2.2.1 LSTM

V literaturi je napogosteje omenjen model, osnovan na besedah z enim nivojem LSTM enot in polno povezano plastjo (glej razdelek 3.1.1). Besede so predstavljene v obliki besednih vložitev.

Poleg osnovne verzije LSTM se pogosto uporablja tudi BLSTM [36], ki se v določenih primerih uči hitreje od LSTM. BLSTM idejo o dvosmernem vhodu iz BRNN (glej razdelek 3.1.3) uporabi na LSTM.

2.2.2 CNN

V delu [39] so naredili pregled nad različnimi nastavitvami hiper parametrov na problemu klasifikacije teksta s CNN. Priporočajo uporabo besednih vložitev, združevalne funkcije *max-pooling* (glej razdelek 3.2.2), konvolucijo po eni dimenziji s filtrom velikosti okoli 10 ali zaporedjem filtrov s skupno velikostjo okoli 10. Pri kompleksnejših modelih predlagajo uporabo izpuščanja nevronov z visoko stopnjo izpuščanja (0.5). Izpuščanje enot (ang. *dropout*) rešuje problem prevelikega prileganja učnim podatkom. Stopnja izpuščanja predstavlja verjetnost izključitve nevrona za trenutni učni primer.

2.2.3 C-LSTM

Članek [40] predlaga novo arhitekturo C-LSTM za klasifikacijo teksta. Glavni komponenti predlagane arhitekture C-LSTM sta plasti CNN in LSTM. Konvolucijska plast iz zaporedja besed izlušči značilnosti, plast LSTM pa poskrbi za pomnjenje dolgoročnih odvisnosti. Predlagana arhitektura dosega rezultate, primerljive uveljavljenim DNN modelom za klasifikacijo teksta.

2.2.4 RCNN

Model RCNN, predlagan v [23], sestavlja plast iz dvosmerne RNN, ki ji sledita konvolucijska plast in združevalna plast z *max-pooling* funkcijo. Z rekurenčnim nivojem se zajame kontekst, konvolucijska plast pa poskrbi za predstavitev podatkov na višjem nivoju. Združitev RNN in CNN daje boljše rezultate od CNN na problemu klasifikacije teksta.

2.3 Uporabljena orodja

2.3.1 Keras

Keras je odprtokodna visokonivojska knjižica za delo z nevronskimi mrežami [19]. Je ovojnica okoli knjižic za delo z nevronskimi mrežami TensorFlow, CNTK in Theano. Razvita je bila z namenom enostavnega in hitrega eksperimentiranja z nevronskimi mrežami. Podpira konvolucijske in rekurenčne tipe nevronskih mrež, ki jih lahko med seboj kombiniramo.

2.3.2 Natural Language Toolkit

NLTK je python knjižica z orodji za delo z naravnim jezikom [27]. Ponuja tokenizacijo in označevanje besedila, različna orodja za statistiko besedila, različne korpuse besedil.

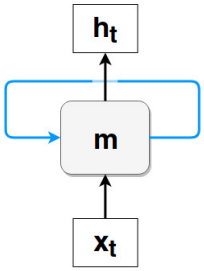
Poglavje 3

Uporabljene metode

3.1 Rekurenčne nevronske mreže (RNN)

Rekurenčne nevronske mreže so razširitev usmerjenih nevronskih mrež, ki vsebujejo ciklične povezave. Cikel omogoča prenos skritega stanja celice v naslednji korak. Zaradi tega se te mreže dobro obnesejo pri problemih, kjer je pomembno ohranjanje informacije skozi zaporedje podatkov. Rekurenčne nevronske mreže se dobro obnesejo pri prepoznavanju govora [10] in rokopisa [11].

Slika 3.1 prikazuje shemo osnovne rekurenčne nevronske mreže iz enega modula. x_t predstavlja vhodno matriko v nevronsko mrežo, h_t izhod iz nevronske mreže, modra povezava pa označuje ciklično povezavo.



Slika 3.1: Shema osnovne RNN (prirejeno po [29]).

Osnovna različica RNN ima problem z rastočim in pojemajočim gradientom.

Pri pojavu rastočega gradienta ima model težave pri učenju kratkoročnih odvisnosti. Odprava rastočega gradienta je relativno enostavna. Problem se lahko odpravi z nastavitvijo uteži < 1 ali omejitvijo največje vrednosti gradienta [31].

Ob pojavu pojemajočega gradienta se model ne nauči dolgoročnih odvisnosti med podatki v daljših zaporedjih [3]. Problem se najpogosteje rešuje z uporabo posebnih aktivacijskih funkcij. Primera takšne rešitve sta LSTM in GRU.

3.1.1 LSTM

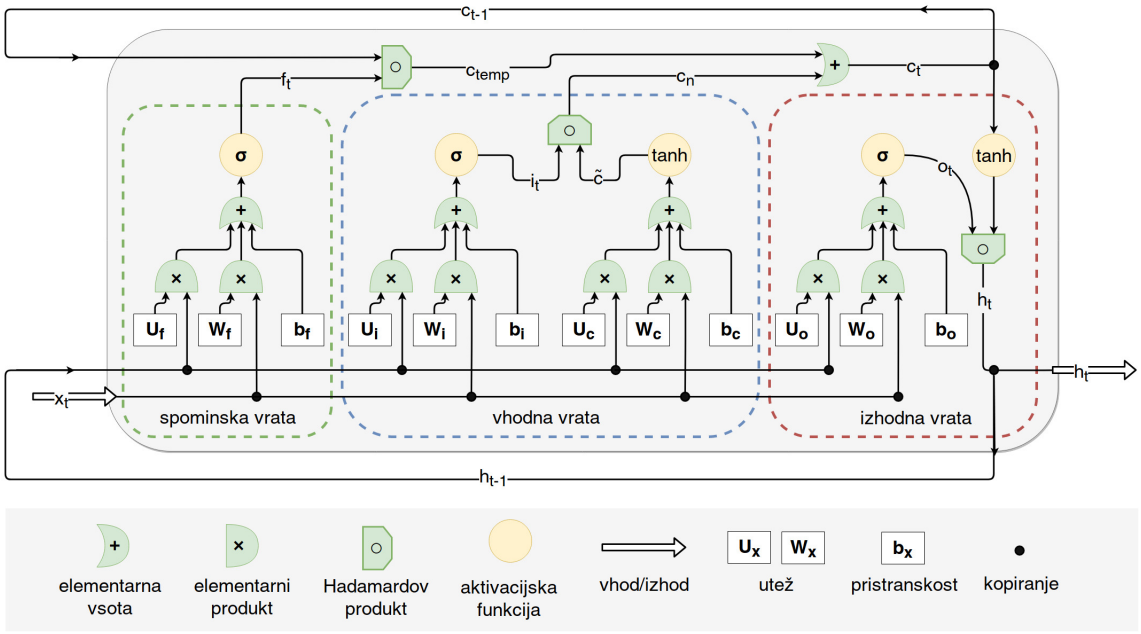
Long Short Term Memory so posebna vrsta RNN, ki se lahko naučijo odvisnosti med podatki v daljših zaporedjih. Učenje dolgoročnih odvisnosti omogoča spominska celica, ki si lahko dolgoročno zapomni majhen del informacije. Na sliki 3.2 je modul LSTM, ki ga sestavlja spominska celica in vrata za izračun vrednosti spominske celice in izhoda.

Spominska celica lahko stanje ob prenosu v naslednji korak ohrani, ga delno spremeni ali popolnoma nadomesti z novim stanjem [29]. Vsebino celice določata spominska in vhodna vrata. Zaradi načina delovanja spominske celice pri učenju z vzratnim razširjanjem napake skozi čas (ang. *backpropagation through time (BPTT)*) ne pride do pojemajočega gradienta, ki se pojavlja pri učenju osnovne RNN.

Spominska vrata določajo, katero informacijo bomo zavrgli iz spominske celice z vsebino c_{t-1}. Izračunana so kot (3.1), kjer je σ_g logistična funkcija, W_f in U_f uteži spominskih vrat, b_f pristranskost spominskih vrat, x_t vhod v LSTM celico in h_{t-1} izhod LSTM celice v prejšnjem koraku.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{3.1}$$

Vmesna vrednost spominske celice (3.2) je izračunana s Hadamardovim produktom med izhodom spominskih vrat f_t (3.1) in prejšnjo vrednostjo



Slika 3.2: Shema LSTM modula (prirejeno po [29]).

spominske celice c_{t-1}. Znak $\circ$ označuje Hadamardov produkt.

$$c_{temp} = f_t \circ c_{t-1} \tag{3.2}$$

Vhodna vrata (3.5) določajo prepustnost nove informacije v spominsko celico. Z omejevanjem prepustnosti spominsko celico ščitijo pred nepomembnimi vhodi. Sestavljena so iz dveh delov. Prvi del (3.3) določa, kateri deli informacije se bodo posodobili. W_i in U_i predstavljata uteži prvega dela vhodnih vrat, b_i pa pristranskost prvega dela vhodnih vrat. Drugi del (3.4) sestavi matriko z novo informacijo. *tanh* predstavlja hiperbolični tangens, W_c in U_c uteži drugega dela vhodnih vrat, b_i pa pristranskost drugega dela vhodnih vrat. Izhod vhodnih vrat (3.5) je združitev obeh delov s Hadamardovim produktom.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3.3}$$

$$\tilde{c} = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3.4}$$

$$c_n = i_t \circ \tilde{c} \tag{3.5}$$

Nova vrednost spominske celice (3.6) je izračunana kot vsota med vmesno vrednostjo spominske celice (3.2) in izhodom vhodnih vrat (3.5).

$$c_t = c_{temp} + c_n \tag{3.6}$$

Izhodna vrata (3.8) določajo prepustnost informacije iz spominske celice. Z omejevanjem prepustnosti preprečujejo posredovanje nepomembnih informacij.

Izhod izhodnih vrat (3.8) se izračuna s Hadamardovim produktom med izhodno prepustnostjo (3.7) in vrednostjo hiperboličnega tangensa nad vrednostjo spominske celice (3.6). W_o in U_o predstavljata uteži izhodnih vrat, b_o pa predstavlja pristranskost izhodnih vrat.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3.7}$$

$$h_t = o_t \circ tanh(c_t) \tag{3.8}$$

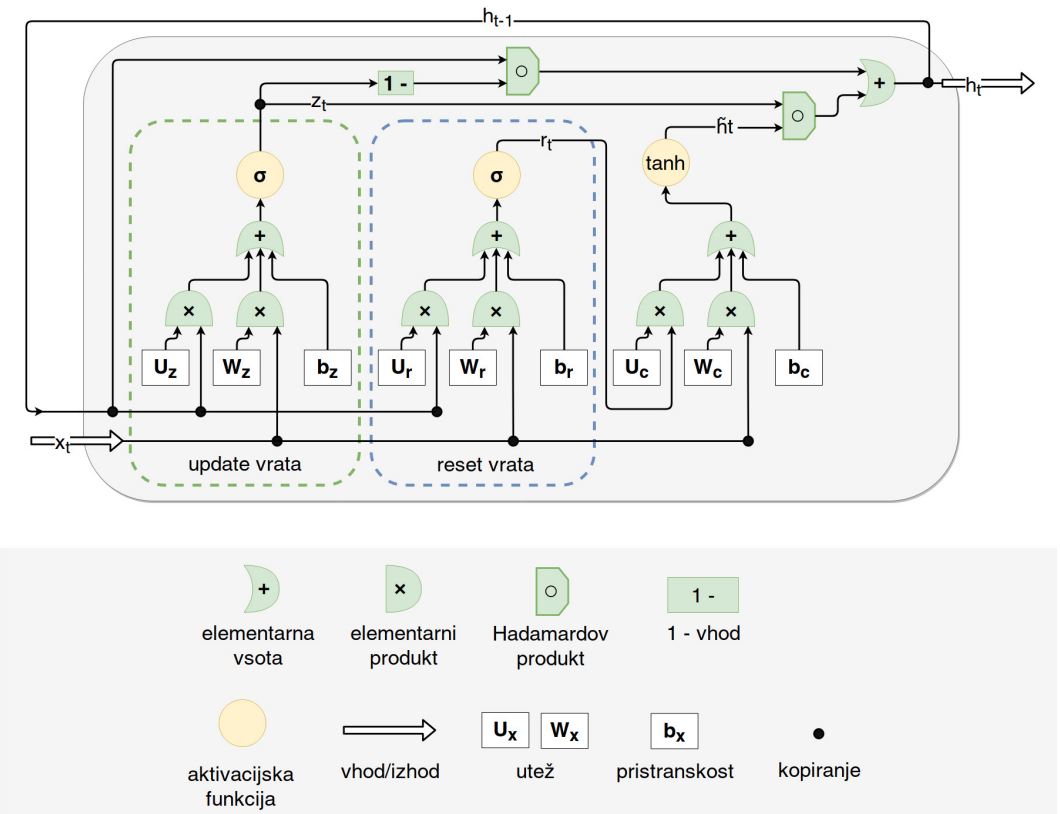
3.1.2 GRU

Modul GRU na sliki 3.3 je sestavljen iz vrat za resetiranje in posodabljanje. Glavna razlika v primerjavi z LSTM je odsotnost ločene spominske celice in izhodnih vrat. Preprostejša arhitektura modula v primerjavi z LSTM pripomore k hitrejšemu učenju in boljšim rezultatom na nekaterih podatkovnih množicah [16].

Izhod GRU (3.9) je izračunan kot vsota med stanjem v prejšnji iteraciji h_{t-1} in predlaganim novim stanjem v trenutnem stanju $\tilde{h_t}$ (3.12). Razmerje med prejšnjim in predlaganemu novemu stanju določa izhod vrat za posodabljanje z_t (3.10). Znak $\circ$ označuje Hadamardov produkt.

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h_t} \tag{3.9}$$

Vrata za posodabljanje (3.10) določajo, koliko informacije iz prejšnjega stanja se prenese v trenutno stanje. Simbol σ_g označuje logistično aktivacijsko funkcijo, W_z in U_z uteži vrat za posodabljanje, b_z pristranskost vrat za



Slika 3.3: Shema GRU modula (prirejeno po [29]).

posodabljanje, x_t vhod v GRU modul in h_{t-1} izhod GRU modula v prejšnjem koraku.

$$z_t = \sigma_g(U_z h_{t-1} + W_z x_t + b_z) \tag{3.10}$$

Vrata za resetiranje (3.11) določajo, katere informacije iz stanja v prejšnji iteraciji naj se zavržejo. Simbola W_r in U_r predstavljata uteži vrat za resetiranje, b_r pa pristranskost vrat za resetiranje.

$$r_t = \sigma_g(U_r h_{t-1} + W_r x_t + b_r) \tag{3.11}$$

Predlagano novo stanje (3.12) je določeno s trenutnim vhodom v modul GRU x_t in Hadamardovim produktom med izhodom vrat za resetiranje r_t

(3.11) in izhodom iz modula v prejšnjem koraku h_{t-1}. Simbola W_c in U_c predstavljata uteži, b_c pa pristranskost predlaganega novega stanja.

$$\tilde{h_t} = tanh(U_c(r_t \circ h_{t-1}) + W_c x_t + b_c) \tag{3.12}$$

3.1.3 Dvosmerne rekurenčne nevronske mreže (BRNN)

BRNN (ang. *Bidirectional RNN*) so razširitev RNN, kjer se vhodno plast z RNN moduli razdeli v dve skupini. Prva skupina modulov prejme vhod v pravem vrstnem redu, druga skupina modulov pa vhod v obratnem vrstnem redu. Skupini se nato združita s povprečenjem istoležnih izhodov [35]. Dodatni vhodi s podatki v obratnem vrstnem redu v nekaterih primerih pospešijo učenje modela in izboljšajo točnost.

3.2 Konvolucijske nevronske mreže (CNN)

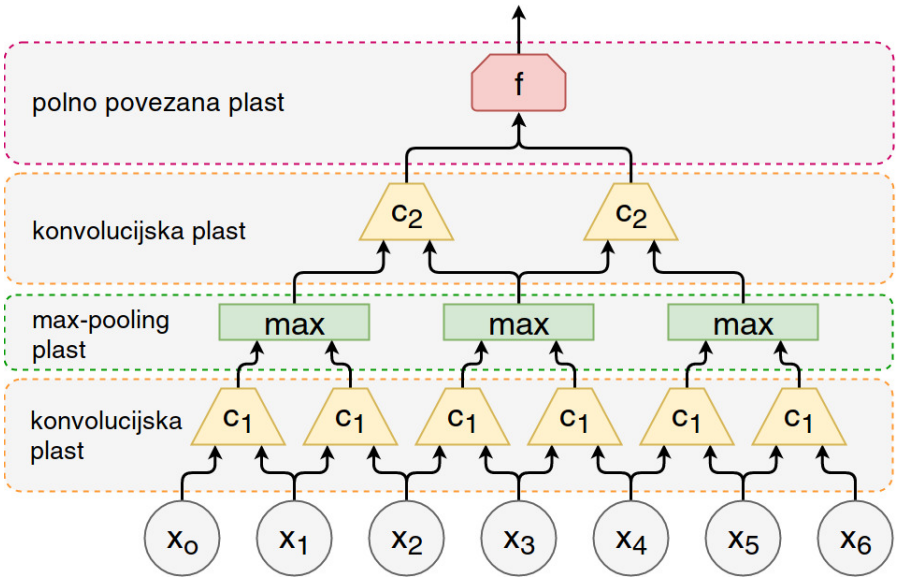
Konvolucijske nevronske mreže so v osnovi namenjene reševanju problemov računalniškega vida. Ideja CNN je, da se s filtri (konvolucijskimi plastmi) določi lokalne značilnosti, ki jih nato združimo v večje skupine (ang. *pooling*), nad katerimi lahko ponovno določamo lokalne značilnosti in jih na koncu uporabimo za napovedovanje v polno povezani plasti [15].

Za reševanje problemov iz procesiranja naravnega jezika se običajno prilagodi operacije konvolucije in združevanja (ang. *pooling*) tako, da delujejo le na eni dimenziji [39].

Na sliki 3.4 je prikazana preprosta shema CNN z dvema konvolucijskima plastema in združevalno plastjo z *max-pooling* funkcijo.

3.2.1 Konvolucijska plast

Konvolucijska plast izvede konvolucijo z naučenim jedrom nad podatki. Enotno jedro za celotno plast omogoča hitrejše učenje mreže.

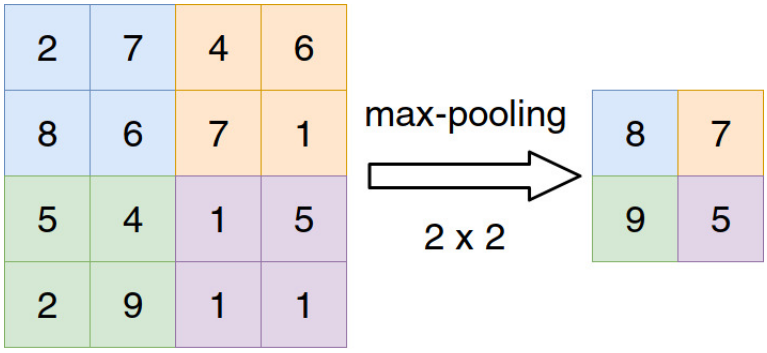


Slika 3.4: Shema preproste CNN (prirejeno po [28]).

3.2.2 Združevalna (ang. *pooling*) plast

Združevalna plast zmanjša število parametrov. Med najbolj popularnimi je metoda *max-pooling*, ki lokalna območja združuje tako, da območja razdeli na neprekrivajoče dele in kot rezultat vrne največjo aktivacijsko vrednost iz posameznega območja. S tem se ohrani informacija o prisotnosti pomembnih značilnosti, a se izgubi njihov položaj [17].

Slika 3.5 prikazuje primer operacije *max-pooling* s filtrom velikosti $2 * 2$. Barve ponazarjajo razdelitev območij na neprekrivajoče se dele velikosti $2 * 2$.



Slika 3.5: *Max-pooling* s filtrom velikosti $2 * 2$.

3.3 Aktivacijske funkcije

3.3.1 Logistična funkcija

Logistična funkcija (3.13) omeji vrednost izhoda na interval med 0 in 1. Slabi lastnosti logistične funkcije sta pojav pojemajočega gradienta in to, da zaloga vrednosti funkcije ni centrirana na 0, kar lahko vodi do počasnejšega učenja [18].

$$f(n) = S(x) = \frac{1}{1 + e^{-x}} \tag{3.13}$$

3.3.2 Hiperbolični tangens

Hiperbolični tangens (3.14) je raztegnjena logistična funkcija, ki omeji vrednost izhoda na interval med -1 in 1. Zaloga vrednosti funkcije je v nasprotju z logistično funkcijo centrirana na 0 [18].

$$f(n) = S(x) = \frac{1}{2 + e^{-2x}} - 1 \tag{3.14}$$

3.3.3 ReLU

Aktivacijska funkcija ReLU (3.15) omeji spodnjo vrednost izhoda na 0. Zaradi preproste matematične operacije lahko v primerjavi z logistično funkcijo

in hiperboličnim tangesom pospeši učenje za faktor 6. Uporaba te aktivacijske funkcije lahko vodi v problem umirajočih nevronov, kjer se uteži nevrona posodobijo tako, da je izhod iz nevrona zaradi uporabe aktivacijske funkcije ReLU za vse vhode enak 0 [18].

$$f(n) = \begin{cases} 0 & \text{za } x < 0 \\ x & \text{za } x \geq 0 \end{cases} \tag{3.15}$$

3.3.4 Softmax

Softmax (3.16) je aktivacijska funkcija, ki vrednosti vhodnega vektorja x skrči na interval med 0 in 1 tako, da je vsota vrednosti enaka 1. Izhod je enak diskretni verjetnostni porazdelitvi [34].

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{N} e^{x_k}} \qquad \forall j \in 1..N \tag{3.16}$$

3.4 Besedne vložitve

Besedne vložitve so ena izmed možnih načinov predstavitve besedil, primerne za učenje nevronskih mrež na problemih procesiranja naravnega jezika. Posamezne besede se preslikajo v vnaprej izračunan vektor realnih števil.

3.4.1 Word2Vec

Word2Vec je orodje za učenje besednih vložitev. Učenje besednih vložitev je možno z dvema različnima modeloma - *continuous bag of words* (CBOW) in *continuous skip-gram* [37].

Učenje besednih vložitev poteka tako, da učimo model za izmišljen problem. Naučenega modela ne uporabljamo za napovedovanje, temveč uporabimo naučene uteži na skritem nivoju, ki predstavljajo vrednosti besednih vložitev.

Pri metodi CBOW se uči model za napovedovanje besede glede na ostale besede v okolici napovedane besede.

Pri metodi *continuous skip-gram* se uči model za obratno nalogo. Uči se model za napovedovanje besed v okolici dane besede. V primerjavi s CBOW je učenje tega modela počasnejše, a daje boljše rezultate pri besedah, ki se redkeje pojavljajo [26].

3.4.2 GloVe

GloVe je algoritem za učenje besednih vložitev. Vrednosti besednih vložitev se izračunajo z gradnjo matrike, ki ponazarja, kako pogosto se določena beseda pojavi v nekem kontekstu. Dimenzije zgrajene matrike se nato zmanjšajo z matrično faktorizacijo. Vrstice v pomanjšani matriki predstavljajo vrednosti besednih vložitev za posamezne besede [32].

3.4.3 FastText

FastText je orodje za klasifikacijo besedil in učenje besednih vložitev [5].

Tako kot *word2vec* tudi *fastText* omogoča učenje besednih vložitev z modeloma continuous bag of words in continuous skip-gram. Glavna razlika v primerjavi z *word2vec* je učenje na zaporedju znakov namesto na besedah. Model se zaradi tega nauči zgradbo besed in lahko zgradi besedne vložitve tudi za besede, ki se ne pojavijo v učnem korpusu [4].

Poglavje 4

Učenje in evalvacija modelov

4.1 Podatkovna množica

Podatkovno množico predstavljeno v tabeli 4.1, smo sestavili iz ročno napisanih in generiranih člankov o hujšanju.

Generirani članki so bili generirani z dvema različnima modeloma, ki sta se razlikovala po kakovosti besedil.

podatkovna množica	**št. člankov**	**št. besed**
ročno napisani članki	18.483	10.209.827
generirani članki s slabim modelom	19.234	11.720.470
generirani članki z dobrim modelom	581	354.644
skupaj	38.298	22.284.941

Tabela 4.1: Sestava podatkovne množice

Podatkovno množico smo razdelili na učno, validacijsko in testno podatkovno množico. Učna množica je sestavljena iz približno 70%, validacijska iz približno 10% in testna množica iz približno 20% celotne podatkovne množice. Podrobna razdelitev podatkovne množice je prikazana v tabeli 4.2.

Učno množico smo uporabili za učenje modela. Validacijsko množico smo uporabili pri izbiri hiper parametrov (dolžina vhoda, tip besedne vložitve,

	število člankov		
podatkovna množica	**učna**	**validacijska**	**testna**
ročno napisani članki	12.938	1.848	3.697
generirani članki s slabim modelom	13.464	1.924	3.847
generirani članki z dobrim modelom	406	59	116
skupaj	26.808	3.831	7.660

Tabela 4.2: Razdelitev podatkovne množice na učno, validacijsko in testno podatkovno množico

število plasti, ...). Testno množico pa smo uporabili za končno primerjavo rezultatov med različnimi modeli.

4.1.1 Normalizacija podatkov

Generirani članki so že bili v normalizirani obliki z besedami, zapisanimi z majhnimi črkami, brez posebnih znakov, ločil in številk.

Ročno napisane članke smo zato pretvorili v enako obliko. Iz člankov smo odstranili vse posebne znake, ločila, številke in pretvorili črke z veliko začetnico v črke z malo začetnico.

Tabela 4.3 prikazuje izseke člankov v normalizirani obliki.

vrsta	primer
ročno napisani	overweight people are routinely told to exercise more however a study in overweight adolescent girls shows that exercise increases free radical stress due to the inefficient utilization of extra oxygen during the exercise process this means that exercise in the presence of a lack of antioxidants will increase tissue damage and consequent inflammation resulting
generirani slab model	general of the question of this is the best way to take the ensuring of exercise you can eat a variety of diet pills and to make it a few the best results that you can do to lose weight you have been able to do it off it is not a few years it will know what you can lose extra weight if you want to do it to stay out and better to get fuller at least days for you to lose weight
generirani dober model	dancing to lose weight is not only fun it can surely help people achieve a slim abs that everyone has the same objective as you wish the aerobics component above is a very unique technique which works very well as a great cardio workout to burn off excess calories and extra fat that effectively stored food in the body when you exercise

Tabela 4.3: Primeri člankov v normalizirani obliki

4.2 Modeli

4.2.1 Znakovni modeli

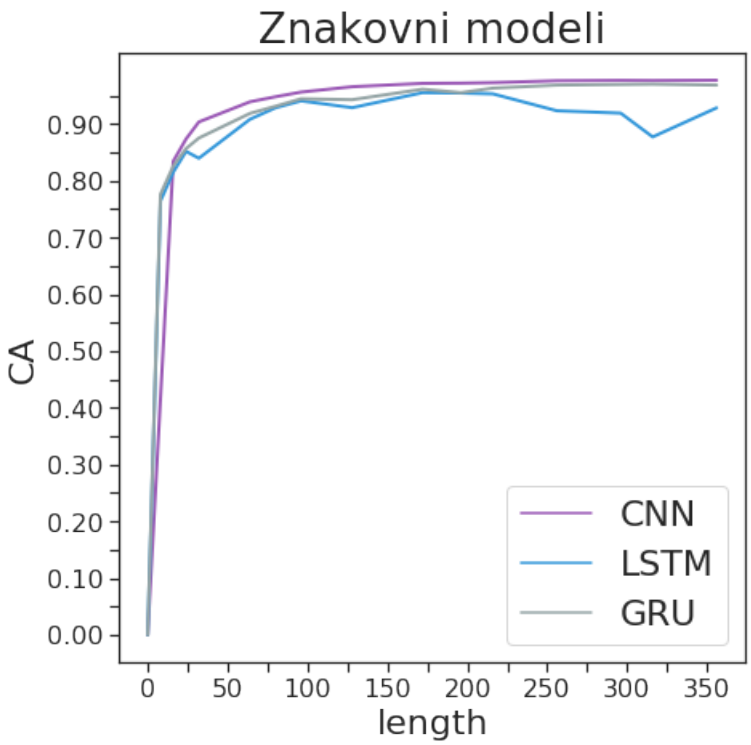
Preizkusili smo tri različne arhitekture modelov za klasifikacijo člankov na podlagi zaporedja znakov v besedilu.

Predstavitev podatkov

Članke smo razdelili na zaporedje znakov določene dolžine. Vsak znak iz zaporedja je imel svoj vhod v nevronsko mrežo.

Znake smo preslikali v *one-hot* vektorje, pri čemer smo nabor znakov omejili na male črke angleške abecede in presledek. Znakovnim zaporedjem, krajšim od določene dolžine, smo na začetek dodali ustrezno število ničel, ki so predstavljale oblogo. Z dodano oblogo smo poenotili dolžino vhodov v model.

Najprimernejšo dolžino vhodov smo določili za vsak model posebej. Na sliki 4.1 je graf, ki prikazuje točnost modela v odvisnosti od dolžine vhoda na validacijski podatkovni množici. Pri modelih CNN in GRU se je klasifikacijska točnost izboljševala sorazmerno z dolžino vhoda. Model LSTM je najboljšo klasifikacijsko točnost dosegel pri dolžini vhoda 172 znakov.



Slika 4.1: Klasifikacijska točnost znakovnih modelov za različne dolžine vhodov na validacijski množici.

Klasifikacija člankov

Modeli za vhod ne sprejmejo celotnega članka, temveč le zaporedje znakov določene dolžine. Za klasifikacijo članka se zato članek razdeli na dele ustrezne dolžine. Za vsak del članka se z modelom napove verjetnost, da je del generiran. Iz vseh napovedi delov članka se nato iz napovedanih verjetnosti izračuna povprečje, ki predstavlja verjetnost, da je članek generiran. Članki s povprečno verjetnostjo enako ali večjo od 50% so klasificirani kot generirani članki, članki s povprečno verjetnostjo manjšo od 50% pa kot ročno napisani članki.

CNN

Znakovni model CNN je sestavljen iz zaporedja konvolucijskih in združevalnih plasti, katerim sledita polno povezani plasti. Tabela 4.4 opisuje podrobno zgradbo modela.

plast	**opis**
1	konvolucijska plast s 128 filtri z jedrom velikosti 5 in aktivacijsko funkcijo *relu*
2	*max-pooling* plast z oknom širine 3
3	konvolucijska plast s 128 filtri z jedrom velikosti 3 in aktivacijsko funkcijo *relu*
4	*max-pooling* plast z oknom širine 3
5	polno povezana plast s 128 nevroni
6	plast s 30% izpuščanjem vhodov
7	polno povezana plast z aktivacijsko funkcijo *softmax*

Tabela 4.4: Plasti znakovnega modela CNN.

Iz matrike zmot v tabeli 4.5 razberemo, da model CNN na testni množici brez napak klasificira ročno napisane članke (100% CA) in članke generirane s slabim modelom (100% CA). Model je vse članke, generirane z dobrim modelom, klasificiral kot ročno napisane članke (0% CA).

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.697	0
	generirani	116 (0 + 116)	3.847 (3.847 + 0)

Tabela 4.5: Matrika zmot znakovnega modela CNN s 356 znaki na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

LSTM

Model LSTM je sestavljen iz nivoja s 16 moduli LSTM s 30% izpuščanjem enot (ang. *dropout*). Nivoju z moduli LSTM sledi polno povezana plast z aktivacijsko funkcijo *softmax*.

Iz matrike zmot v tabeli 4.6 razberemo, da model LSTM s 172 znaki na testni podatkovni množici brez napak klasificira ročno napisane članke (100% CA) in članke, generirane s slabim modelom (100% CA). Model člankov, generiranih z dobrim modelom, ni zaznal kot generirane (0% CA).

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.697	0
	generirani	116 (0 + 116)	3.847 (3.847 + 0)

Tabela 4.6: Matrika zmot znakovnega modela LSTM s 172 znaki na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

GRU

Model GRU je sestavljen iz nivoja z 32 moduli GRU s 30% izpuščanjem enot. Nivoju GRU sledi polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.

Iz matrike zmot v tabeli 4.7 razberemo, da je model na testni množici brez napak klasificiral ročno napisane članke (100% CA) in članke, generirane s slabim modelom (100% CA). Model člankov, generiranih z dobrim modelom, ni zaznal kot generirane (0% CA).

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.697	0
	generirani	116 (0 + 116)	3.847 (3.847 + 0)

Tabela 4.7: Matrika zmot znakovnega modela GRU s 316 znaki na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

4.2.2 Besedni modeli

Preizkusili smo šest različnih arhitektur DNN za klasifikacijo člankov na podlagi zaporedja besed. Pri vsakem modelu smo preizkusili točnost modela z različnimi vnaprej naučenimi besednimi vložitvami.

Predstavitev podatkov

Članke smo razdelili na zaporedja besed dolžine 128. Besede smo nato preslikali v vektorje besednih vložitev in iz njih sestavili matriko dimenzij

*dolžina zaporedja * dolžina vektorja besedne vložitve*

Preizkusili smo tri različne vnaprej naučene besedne vložitve:

Word2vec

300 dimenzionalni vektorji besednih vložitev, naučeni na podatkovni množici Google Newsl [9].

GloVe

300 dimenzionalni vektorji besednih vložitev, naučeni na podatkovni množici Common Crawl [7].

FastText

300 dimenzionalni vektorji besednih vložitev, naučeni na podatkovni množici Wikipedia [6].

Klasifikacija člankov

Modeli za vhod ne sprejmejo celotnega članka, temveč zaporedje 128 besed. Za klasifikacijo članka se zato članek razdeli na dele dolžine 128 besed. Za vsak del članka se z modelom napove verjetnost, da je del generiran. Iz vseh napovedi delov članka se nato iz napovedanih verjetnosti izračuna povprečje, ki predstavlja verjetnost, da je članek generiran. Članki s povprečno verjetnostjo enako ali večjo od 50% so klasificirani kot generirani članki, članki s povprečno verjetnostjo manjšo od 50% pa kot ročno napisani članki.

Modeli so dele člankov, generiranih z dobrim modelom, zaradi majhnega števila primerov, pogosto klasificirali kot ročno napisane. Napovedane verjetnosti pri napačno klasificiranih člankih, generiranih z dobrim modelom, so bile precej višje od pravilno klasificiranih ročno napisanih člankov. Klasifikacijsko točnost modelov pri napovedi delov člankov za članke, generirane z dobrim modelom, smo zato izboljšali z množenjem napovedanih verjetnosti s faktorjem, ki smo ga empirično določili na validacijski podatkovni množici. S tem smo izboljšali tudi klasifikacijsko točnost pri klasifikaciji celotnih člankov.

CNN

Model CNN sestavlja zaporedje konvolucijskih in združevalnih plasti. Tabela 4.8 opisuje podrobno zgradbo besednega modela CNN.

Graf 4.2 prikazuje primerjavo med besednimi vložitvami na validacijski podatkovni množici. Med besednimi vložitvami so se največje razlike pokazale pri klasifikaciji ročno napisanih člankov. Najbolje je članke ločeval model z uporabo besednih vložitev *word2vec*. Model z uporabo besednih vložitev *gloVe* je dosegel 100% CA na validacijski podatkovni množici pri obeh množicah generiranih člankov, a dosegel zelo slabe rezultate pri ročno napisanih člankih (61,61% CA). Z uporabo besednih vložitev *fastText* je

plast	opis
1	konvolucijska plast s 64 filtri z jedrom velikosti 7 in aktivacijsko funkcijo *relu*
2	*max-pooling* plast z oknom širine 2
3	konvolucijska plast s 64 filtri z jedrom velikosti 5 in aktivacijsko funkcijo *relu*
4	*max-pooling* plast z oknom širine 2
5	konvolucijska plast s 64 filtri z jedrom velikosti 2 in aktivacijsko funkcijo *relu*
6	globalna *max-pooling* plast
7	polno povezana plast z aktivacijsko funkcijo *softmax*

Tabela 4.8: Plasti besednega modela CNN.

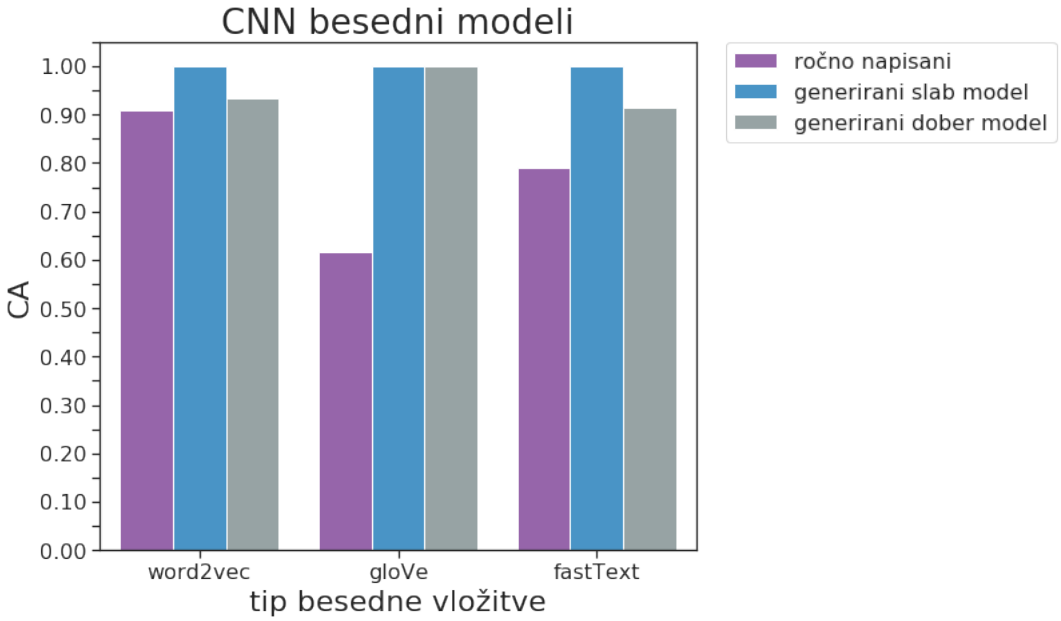
model v primerjavi z besednimi vložitvami *gloVe* dosegel boljše rezultate na ročno napisanih člankih (78, 98% CA) in slabše rezultate na člankih, generiranih z dobrim modelom (91, 52% CA).

Tabela 4.9 z matriko zmot podrobno prikazuje napovedi najboljšega modela CNN z besednimi vložitvami *word2vec* na testni podatkovni množici. Model je na testni podatkovni množici dosegel 91, 60% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 93, 10% CA pri člankih, generiranih z dobrim modelom.

GRU

Model GRU je sestavljen iz nivoja s 64 moduli GRU s 30% izpuščanjem enot. Nivoju GRU sledi polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.

Graf 4.3 prikazuje primerjavo med besednimi vložitvami. Med besednimi vložitvami ni večjih razlik pri napovedovanju ročno napisanih člankov in člankov, generiranih s slabim modelom. Razlike med besednimi vložitvami se pojavijo pri klasifikaciji člankov, generiranih z dobrim modelom. Tu daje



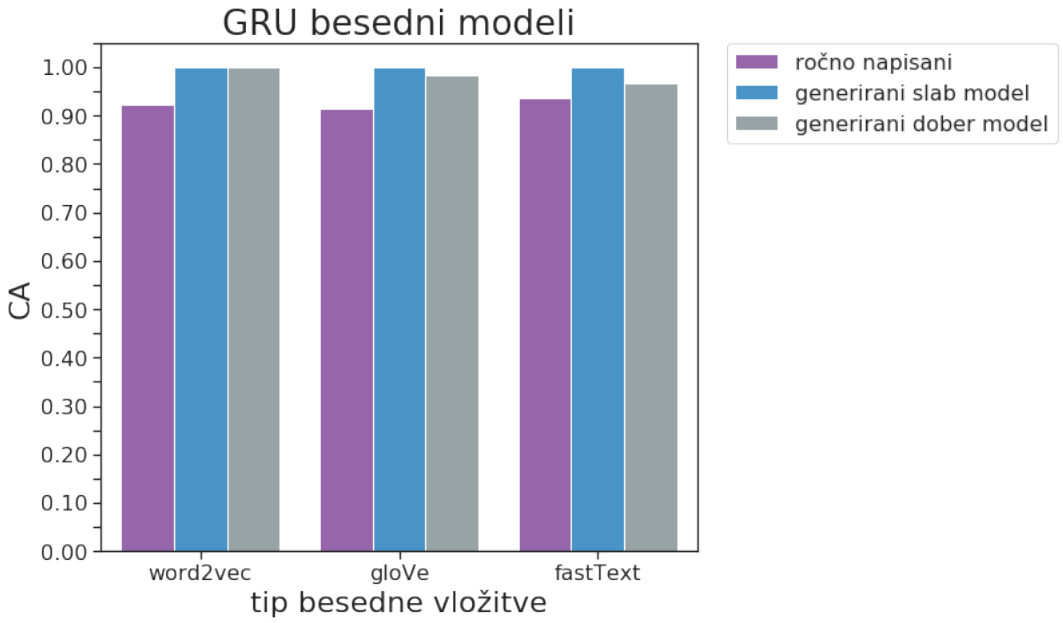
Slika 4.2: Primerjava klasifikacijske točnosti besednega modela CNN z različnimi besednimi vložitvami na validacijski podatkovni množici.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.371	309
	generirani	8 (0 + 8)	3.932 (3.824 + 108)

Tabela 4.9: Matrika zmot besednega modela CNN z besednimi vložitvami *word2vec* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

najboljše napovedi besedna vložitev *word2vec*, pri kateri model doseže 100% klasifikacijsko točnost na validacijski podatkovni množici. Modela z bese-

dnima vložitvama *gloVe* in *fastText* sta pri klasifikaciji člankov, generiranih z dobrim modelom na validacijski podatkovni množici dosegla 98,31% CA (*gloVe*) in 96,61% CA (*fastText*).



Slika 4.3: Primerjava klasifikacijske točnosti besednega modela GRU pri dolžini vhoda 128 besed z različnimi besednimi vložitvami na validacijski podatkovni množici.

Matrika zmot v tabeli 4.10 podrobno prikazuje napovedi najboljšega modela GRU z besednimi vložitvami *word2vec* na testni podatkovni množici. Model je na testni podatkovni množici dosegel 92,85% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 96,55% CA pri člankih, generiranih z dobrim modelom.

LSTM

Model LSTM je sestavljen iz nivoja s 64 moduli LSTM s 30% izpuščanjem enot. Nivoju LSTM sledi polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.550	130
	generirani	12 (0 + 12)	3928 (3824 + 104)

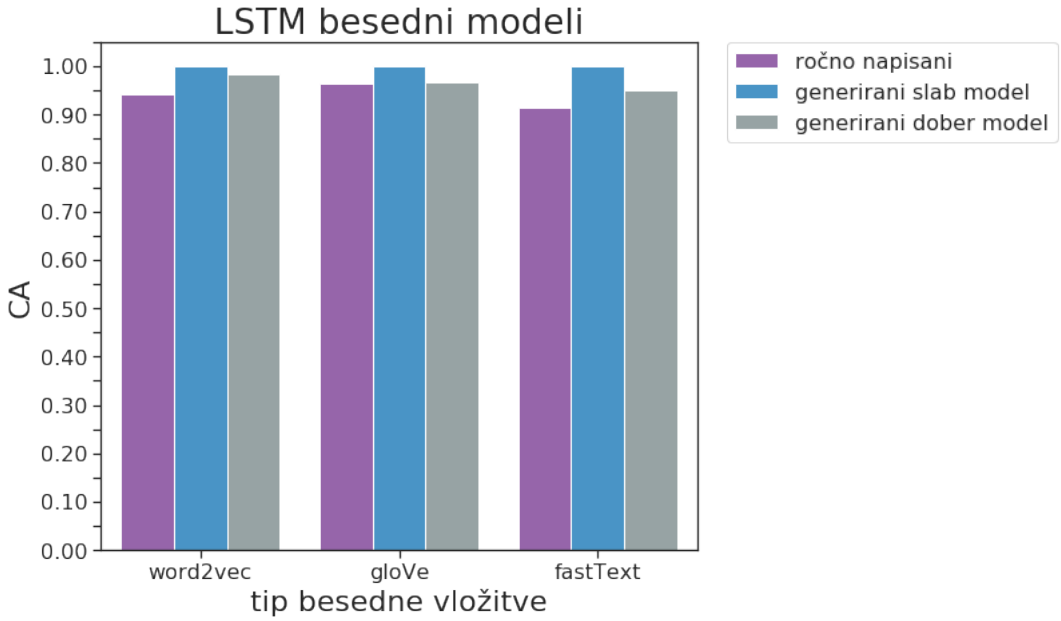
Tabela 4.10: Matrika zmot besednega modela GRU z besednimi vložitvami *word2vec* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

Graf na sliki 4.4 prikazuje primerjavo med besednimi vložitvami na validacijski podatkovni množici. Vsi modeli brez napak ločijo članke, generirane s slabim modelom. Najboljše rezultate dosega besedna vložitev *gloVe* (96, 46% CA pri ročno napisanih člankih in 96, 61% CA pri člankih, generiranih z dobrim modelom na validacijski podatkovni množici). Slabše se pri klasifikaciji ročno napisanih člankov obneseta modela z uporabo besedne vložitve *word2vec* (94, 20% CA na validacijski podatkovni množici) in *fastText* (91, 42% CA na validacijski podatkovni množici).

V tabeli 4.11 so z matriko zmot podrobno prikazane napovedi najuspešnejšega modela LSTM z besednimi vložitvami *gloVe* na testni podatkovni množici. Model je na testni podatkovni množici dosegel 96, 30% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 91, 37% CA pri člankih, generiranih z dobrim modelom.

BLSTM

Model BLSTM je sestavljen iz nivoja s 64 moduli BLSTM s 30% izpuščanjem enot. Nivoju BLSTM sledi polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.



Slika 4.4: Primerjava klasifikacijske točnosti besednega modela LSTM z različnimi besednimi vložitvami na validacijski podatkovni množici.

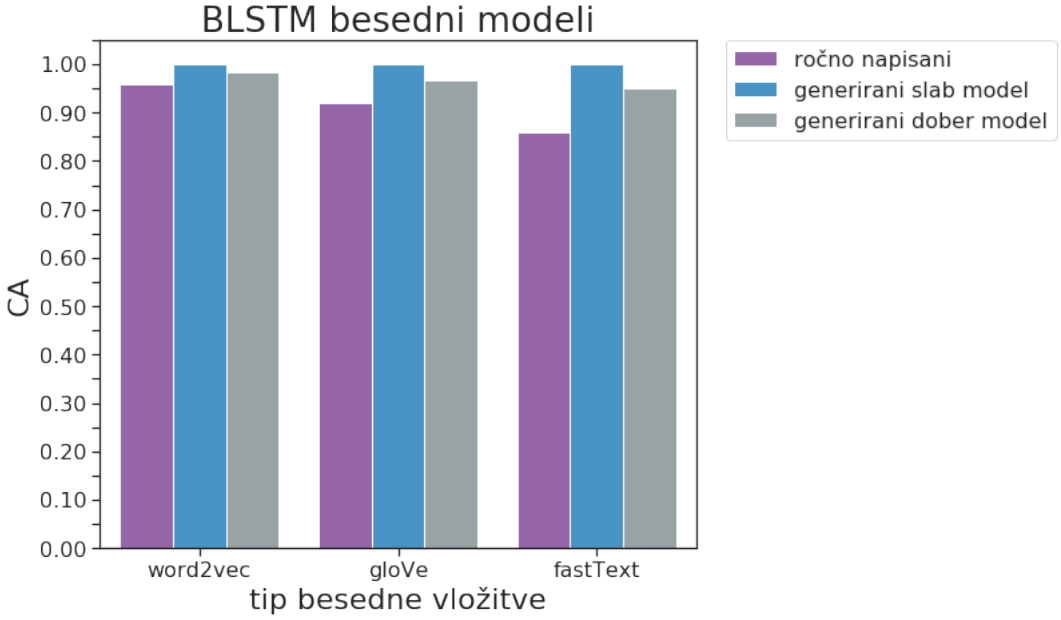
		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.544	136
	generirani	10 (0 + 10)	3.930 (3.824 + 106)

Tabela 4.11: Matrika zmot besednega modela LSTM z besednimi vložitvami *gloVe* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

Graf 4.4 prikazuje primerjavo med besednimi vložitvami na validacijski podatkovni množici. Najboljše rezultate je dosegel model z uporabo besednih

vložitev *word2vec* (95, 71% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 98, 30% CA pri člankih, generiranih z dobrim modelom na validacijski podatkovni množici). Vsi modeli so brez napak klasificirali članke, generirane s slabim modelom. Najslabše rezultate smo dosegli z uporabo besednih vložitev *fastText*.

Tabela 4.12 z matriko zmot prikazuje rezultate modela z uporabo besednih vložitev *word2vec* na testni podatkovni množici. Model je na testni podatkovni množici dosegel 96, 71% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 97, 41% CA pri člankih, generiranih z dobrim modelom.



Slika 4.5: Primerjava klasifikacijske točnosti besednega modela BLSTM z različnimi besednimi vložitvami na validacijski podatkovni množici.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.559	121
	generirani	3 (0 + 3)	3.937 (3.824 + 113)

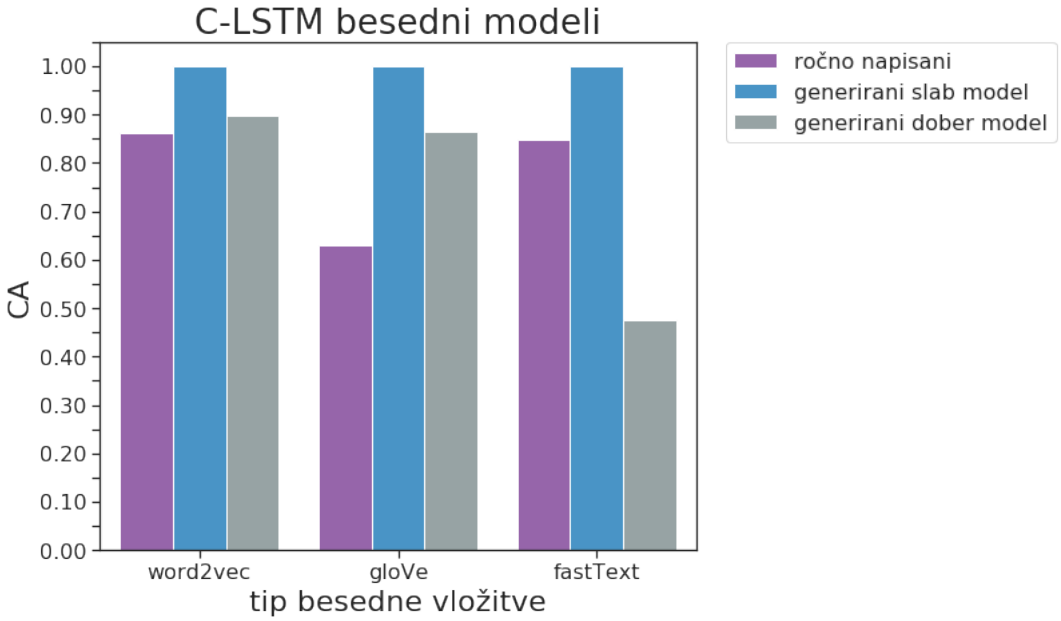
Tabela 4.12: Matrika zmot besednega modela BLSTM z besednimi vložitvami *word2vec* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

C-LSTM

Model C-LSTM je sestavljen iz kombinacije CNN in LSTM nivojev. Model sestavlja konvolucijska plast s 64 filtri z jedrom velikosti 7 in aktivacijsko funkcijo *relu*. Sledi ji plast LSTM s 64 moduli ter 30% izpuščanjem enot. CNN in LSTM nivojema sledi polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.

Iz grafa s primerjavo med besednimi vložitvami na sliki 4.6 je razvidno, da je model pri vseh besednih vložitvah brez napak ločeval članke, generirane s slabim modelom. Model je najboljše rezultate dosegel z uporabo besednih vložitev *word2vec*. Besedni vložitvi *gloVe* in *fastText* sta dosegali veliko slabše rezultate. Z uporabo besednih vložitev *gloVe* je model zelo slabo klasificiral ročno napisane članke ($63,00\%$ CA). Pri besednih vložitvah *fastText* pa je model slabo klasificiral članke, generirane z dobrim modelom ($47,45\%$ CA).

V tabeli 4.13 z matriko zmot so prikazani rezultati najboljšega modela C-LSTM z besedno vložitvijo *word2vec* na testni podatkovni množici. Model je na testni podatkovni množici dosegel $88,17\%$ CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in $93,10\%$ CA pri člankih, generiranih z dobrim modelom.



Slika 4.6: Primerjava klasifikacijske točnosti besednega modela C-LSTM pri dolžini vhoda 128 besed z različnimi besednimi vložitvami na validacijski podatkovni množici.

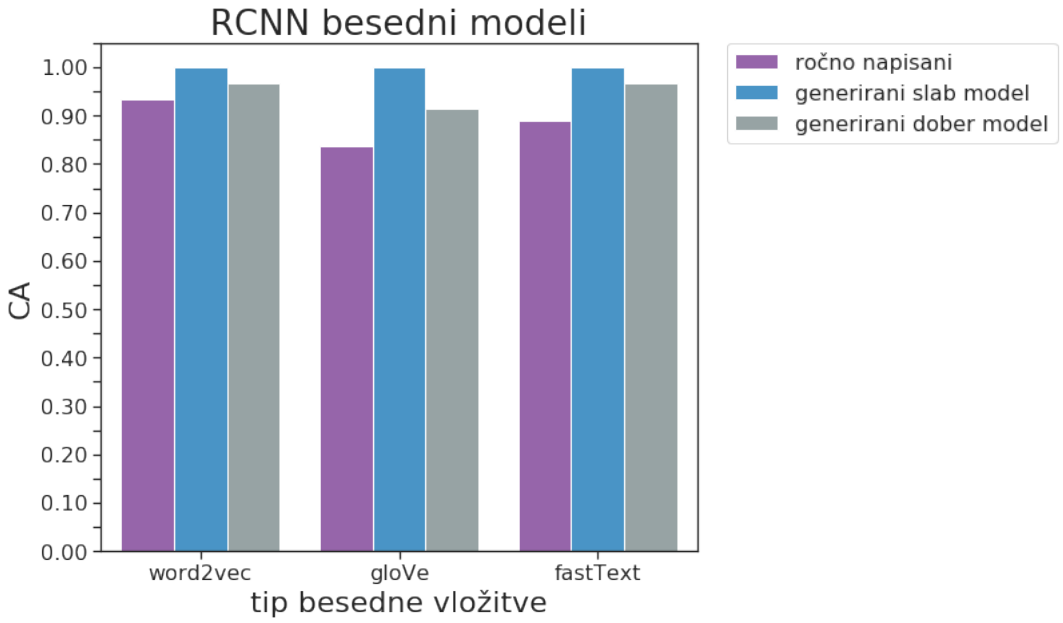
		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.245	435
	generirani	8 (0 + 8)	3.932 (3.824 + 108)

Tabela 4.13: Matrika zmot besednega modela C-LSTM z besednimi vložitvami *word2vec* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

RCNN

Model RCNN sestavlja plast iz 64 dvosmernih modulov RNN, ki ji sledi konvolucijska plast z aktivacijsko funkcijo *relu* s 64 filtri z jedrom velikosti 2 in plast z globalno *max-pooling* operacijo. Zadnja plast modela je polno povezana plast z aktivacijsko funkcijo *softmax* z dvema izhodoma.

Iz grafa na sliki 4.7 je razvidno, da je model dosegel najboljše rezultate z uporabo besednih vložitev *word2vec* (93, 45% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 96, 61% CA pri člankih, generiranih z dobrim modelom). Z uporabo besednih vložitev *fastText* je model slabše klasificiral ročno napisane članke. Z besednimi vložitvami *gloVe* je model v primerjavi z besednimi vložitvami *word2vec* dosegel slabše rezultate na ročno napisanih člankih in člankih, generiranih z dobrim modelom.



Slika 4.7: Primerjava klasifikacijske točnosti besednega modela RCNN z različnimi besednimi vložitvami na validacijski podatkovni množici.

Z matriko zmot v tabeli 4.14 so prikazani rezultati modela z besedno

vložitvijo *word2vec* na testni podatkovni množici. Model je na testni podatkovni množici dosegel 93, 88% CA pri ročno napisanih člankih, 100% CA pri člankih, generiranih s slabim modelom in 94, 82% CA pri člankih, generiranih z dobrim modelom.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	3.455	225
	generirani	6 (0 + 6)	3.934 (3.824 + 110)

Tabela 4.14: Matrika zmot besednega modela RCNN z besednimi vložitvami *word2vec* na testni podatkovni množici. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

4.3 Časi učenja in klasifikacije

Pri vseh modelih smo merili CPU čas učenja in klasifikacije.

4.3.1 Opis sistema

Časi modelov so bili merjeni na spodaj opisanem sistemu:

CPU Intel i5 3570k @4.2Ghz

GPU Nvidia GeForce GTX 1070, gonilnik nvidia 414

OS Manjaro 18, kernel Linux 4.17.19-1

CUDA 10.0.130

CUDNN 7.3.0-1

Anaconda 5.2.0

Python 3.6.5

Keras 2.2.2

Tensorflow 1.9.0

4.3.2 Empirično zmerjeni časi učenja in klasifikacije

Tabela 4.15 prikazuje CPU čase učenja modelov na učni podatkovni množici in CPU čase klasifikacije enega članka s 546 besedami (3132 znaki). Največji vpliv na čas sta imela izbira tipa nevronske mreže (CNN ali RNN) in dolžina vhoda. Modeli, osnovani na arhitekturi CNN, so bili v primerjavi z RNN tudi do 12× hitrejši. Pri modelih RNN se je za najhitrejšega izkazal model z arhitekturo GRU.

model	**opis**	**CPU čas učenja**	**CPU čas klasifikacije**
CNN	356 znakov	36 min 1 s	1,5 ms
LSTM	172 znakov	3 h 11 min 33 s	3,7 ms
GRU	316 znakov	7 h 45 min	4,9 ms
CNN	128 besed, besedna vložitev word2vec	18 min 16 s	0,8 ms
GRU	128 besed, besedna vložitev gloVe	1 h 41 min 35 s	12,6 ms
LSTM	128 besed, besedna vložitev word2vec	2 h 6 min 2 s	15,7 ms
BLSTM	128 besed, besedna vložitev word2vec	2 h 2 min 53 s	31,1 ms
C-LSTM	128 besed, besedna vložitev word2vec	2 h 1 min 19 s	15,7 ms
RCNN	128 besed, besedna vložitev word2vec	1 h 28 min 1 s	6,6 ms

Tabela 4.15: CPU časi učenja modelov in klasifikacije enega primera.

4.4 Rezultati

Vsi modeli so brez napak klasificirali članke iz množice člankov, generiranih s slabim modelom. Pri klasifikaciji člankov, generiranih z dobrim modelom, so bili uspešni le besedni modeli. Razlog za to je najverjetneje majhno število člankov, generiranih z dobrim modelom, v primerjavi z ročno napisanimi članki in članki, generiranimi s slabim modelom.

Znakovni modeli so brez napak klasificirali ročno napisane članke in članke, generirane s slabim modelom. Članke, generirane z dobrim modelom, so vsi znakovni modeli klasificirali napačno.

Razlike med znakovnimi modeli so bile zelo majhne tudi pred klasifikacijo celotnih člankov. V tabeli 5.2 v dodatku A so z matriko zmot prikazani

rezultati znakovnega modela LSTM, ki je edini izmed znakovnih modelov pri klasifikaciji delov člankov dvakrat pravilno klasificiral del članka iz množice člankov, generiranih z dobrim modelom. Matrika zmot v tabeli 5.1 v dodatku A prikazuje rezultate znakovnega modela CNN pri klasifikaciji delov člankov na testni podatkovni množici. Ta model je dosegel najboljšo klasifikacijsko točnost pri klasifikaciji delov člankov med znakovnimi modeli na ročno napisanih člankih. Znakovni model GRU je dosegel najboljše rezultate med znakovnimi modeli pri klasifikaciji člankov, generiranih s slabim modelom. Podrobni rezultati znakovnega modela GRU pri klasifikaciji delov člankov so prikazani z matriko zmot v tabeli 5.3 v dodatku A.

Besedni modeli so bili pri klasificiranju člankov, generiranih z dobrim modelom, uspešnejši. Pri uspešnosti modela je imela velik vpliv izbira besedne vložitve. Med besednimi vložitvami so se na isti arhitekturi nevronske mreže pojavile velike razlike v točnosti modelov. Izbira najboljše besedne vložitve je bila zelo odvisna od modela. Najboljše rezultate smo največkrat dosegli z uporabo besednih vložitev *word2vec*.

Z RNN smo najboljše rezultate na testni podatkovni množici dosegli z modelom BLSTM z uporabo besednih vložitev *word2vec*, ki je dosegel najboljše rezultate med vsemi preizkušenimi modeli. Model GRU je dosegal malo boljše rezultate od LSTM brez dvosmernega vhoda. Model s CNN se je izkazal za slabšega od modelov z RNN. Model RCNN, ki je sestavljen z združenjem arhitektur RNN in CNN, je na testni podatkovni množici dosegel rezultate, primerljive z ostalimi modeli RNN. Zanimivo je, da smo pri arhitekturi C-LSTM, ki združuje arhitekturi RNN in CNN v obratnem vrstnem redu, dosegli najslabše rezultate med besednimi modeli.

Pri klasifikaciji delov člankov je najboljše rezultate tako kot pri klasifikaciji celotnih člankov dosegel model BLSTM. Tabela 5.7 v dodatku B z matriko zmot prikazuje rezultate besednega modela BLSTM pri klasifikaciji delov člankov. Pri člankih, generiranih s slabim modelom sta boljše rezultate od modela BLSTM pri klasifikaciji delov člankov dosegla modela C-LSTM (rezultati z matriko zmot so prikazani v tabeli 5.8 v dodatku B) in RCNN

(rezultati z matriko zmot so prikazani v tabeli 5.9 v dodatku B), ki sta pri člankih, generiranih s slabim modelom dosegla 99,99% CA. Besedni model LSTM je pri klasifikaciji delov člankov dosegel najboljše rezultate izmed vseh besednih modelov pri klasifikaciji ročno napisanih člankov. Rezultati tega modela pri klasifikaciji delov člankov so prikazani z matriko zmot v tabeli 5.5 v dodatku B. Besedni model GRU je pri klasifikaciji delov člankov iz množice člankov, generiranih z dobrim modelom, dosegel 89,69% CA (rezultati z matriko zmot so prikazani v tabeli 5.6 v dodatku B) in zaostal samo za besednim modelom BLSTM. Besedni model CNN je dosegel precejslabše rezultate od modelov RNN pri klasifikaciji delov člankov na člankih, generiranih z dobrim modelom. Tabela 5.4 v dodatku B z matriko zmot prikazuje rezultate besednega modela CNN pri klasifikaciji delov člankov.

V tabeli 4.16 je prikazana klasifikacijska točnost vseh modelov na testni podatkovni množici. Tabela 5.10 v dodatku C prikazuje klasifikacijsko točnost vseh modelov na testni podatkovni množici pri klasifikaciji delov člankov.

model	**opis**	**r**	**g1**	**g2**
CNN	356 znakov	1,0000	1,0000	0,0000
GRU	316 znakov	1,0000	1,0000	0,0000
LSTM	172 znakov	1,0000	1,0000	0,0000
CNN	besedna vložitev word2vec	0,9160	1,0000	0,9310
GRU	besedna vložitev word2vec	0,9285	1,0000	0,9655
LSTM	besedna vložitev gloVe	0,9630	1,0000	0,9137
BLSTM	**besedna vložitev word2vec**	**0,9671**	**1,0000**	**0,9741**
C-LSTM	besedna vložitev word2vec	0,8817	1,0000	0,9310
RCNN	besedna vložitev word2vec	0,9388	1,0000	0,9482

Tabela 4.16: Klasifikacijska točnost vseh modelov na testni podatkovni množici, razdeljeni po vrsti člankov, kjer so r ročno napisani članki, g1 članki, generirani s slabim modelom in g2 članki, generirani z dobrim modelom.

Poglavje 5

Zaključek

V diplomskem delu smo na problemu ločevanja ročno napisanih in avtomatsko generiranih člankov preizkusili različne arhitekture globokih nevronskih mrež in različne predstavitve za vhod v mrežo. Uporaba znakovne predstavitve se je izkazala za dovolj dobro na lažjem problemu ob dovolj veliki učni množici. Predstavitev podatkov z besednimi vložitvami se je izkazala za boljšo od znakovne predstavitve. Za dosego dobrih rezultatov je bila pri tem zelo pomembna izbira vrste besednih vložitev. Pri isti arhitekturi nevronske mreže so se pri uporabi različnih besednih vložitev pojavile velike razlike v točnosti modelov.

Vsi preizkušeni modeli so brez napak klasificirali članke, generirane s slabim modelom. Znakovni modeli niso uspeli pravilno klasificirati člankov, generiranih z dobrim modelom. Besedni modeli so v nasprotju z znakovnimi modeli uspešno klasificirali tudi članke, generirane z dobrim modelom. Najboljše rezultate smo dosegli z besednim modelom, z uporabo arhitekture BLSTM in besedne vložitve *word2vec*. Model z uporabo arhitekture BLSTM je dosegel $96,71\%$ CA pri klasifikaciji ročno napisanih člankov, 100% CA pri klasifikaciji člankov, generiranih s slabim modelom in $97,41\%$ CA pri klasifikaciji člankov, generiranih z dobrim modelom na testni podatkovni množici. Rezultati vseh preizkušenih modelov so prikazani v tabeli 4.16.

V nadaljnjem delu bi bilo potrebno preizkusiti še večji nabor različnih

modelov, vendar bi bilo za bolj zanesljive rezultate potrebno povečati število člankov, zgeneriranih z dobrim modelom. V delu smo se omejili le na članke s tematiko hujšanja. Zanimivo bi bilo raziskati, kako dobro se preizkušeni modeli obnesejo na splošnejši podatkovni množici. Poleg uporabe prednaučenih besednih vložitev bi bilo vredno poizkusiti, kako dobro se obnesejo besedne vložitve, naučene na lastni podatkovni množici.

Dodatek A
Tabele znakovnih modelov

CNN

Tabela 5.1 prikazuje matriko zmot modela CNN pri klasifikaciji delov člankov iz testne podatkovne množice, kjer so bili članki razdeljeni v dele dolžine 356 znakov.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	165.593	497
	generirani	6.219 (565 + 5.654)	123.872 (123.872 + 0)

Tabela 5.1: Matrika zmot znakovnega modela CNN s 356 znaki na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

LSTM

Tabela 5.2 prikazuje matriko zmot modela LSTM pri klasifikaciji delov člankov iz testne podatkovne množice, kjer so bili članki razdeljeni v dele dolžine 172 znakov.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	172.703	3.988
	generirani	7658 (2003 + 5655)	133435 (133433 + 2)

Tabela 5.2: Matrika zmot znakovnega modela LSTM s 172 znaki na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

GRU

Tabela 5.3 prikazuje matriko zmot modela GRU pri klasifikaciji delov člankov iz testne podatkovne množice, kjer so bili članki razdeljeni v dele dolžine 316 znakov.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	168.616	3.221
	generirani	6.158 (503 + 5.655)	129.928 (129.928 + 0)

Tabela 5.3: Matrika zmot znakovnega modela GRU s 316 znaki na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

Dodatek B
Tabele besednih modelov

CNN

Tabela 5.4 z matriko zmot prikazuje napovedi modela CNN pri napovedovanju delov člankov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	14.534	2.220
	generirani	99 (5 + 94)	15.311 (14.939 + 372)

Tabela 5.4: Matrika zmot besednega modela CNN s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

LSTM

Tabela 5.5 z matriko zmot prikazuje napovedi modela LSTM pri napovedovanju delov člankov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	15.432	1.322
	generirani	79 (6 + 73)	15.331 (14.938 + 393)

Tabela 5.5: Matrika zmot besednega modela LSTM s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

GRU

Tabela 5.6 z matriko zmot prikazuje napovedi modela GRU pri napovedovanju delov člankov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	14.835	1.919
	generirani	51 (3 + 48)	15.359 (14.941 + 418)

Tabela 5.6: Matrika zmot besednega modela GRU s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

BLSTM

Tabela 5.7 z matriko zmot prikazuje napovedi modela BLSTM pri napovedovanju delov člankov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	14.879	1.875
	generirani	38 (3 + 35)	15.372 (14.941 + 431)

Tabela 5.7: Matrika zmot besednega modela BLSTM s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

C-LSTM

Tabela 5.8 z matriko zmot prikazuje napovedi modela C-LSTM pri napovedovanju delov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	13.632	3.122
	generirani	103 (1 + 102)	15.307 (14.943 + 364)

Tabela 5.8: Matrika zmot besednega modela C-LSTM s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

RCNN

Tabela 5.9 z matriko zmot prikazuje napovedi modela RCNN pri napovedovanju delov člankov iz testne podatkovne množice, kjer smo članke razdelili na dele dolžine 128 besed.

		napovedani razred	
		ročno	generirani
resnični razred	ročno	14.594	2.160
	generirani	64 (1 + 63)	15.346 (14.943 + 403)

Tabela 5.9: Matrika zmot besednega modela RCNN s 128 besedami na testni podatkovni množici pri klasifikaciji delov člankov. V oklepajih so prikazane napovedi na podatkovnih množicah s slabimi (na prvem mestu) in dobro generiranimi članki (na drugem mestu vsote).

Dodatek C
Rezultati modelov pri klasifikaciji delov člankov

Tabela 5.10 prikazuje rezultate vseh modelov pri klasifikaciji delov člankov pred povprečenjem napovedi za klasifikacijo celotnega članka.

model	**opis**	**r**	**g1**	**g2**
CNN	356 znakov	**0,9970**	0, 9955	0, 0000
GRU	316 znakov	0, 9813	**0,9961**	0, 0000
LSTM	172 znakov	0, 9774	0, 9852	**0,0003**
CNN	besedna vložitev word2vec	0, 8674	0, 9996	0, 7982
GRU	besedna vložitev word2vec	0, 8854	0, 9997	0, 8969
LSTM	besedna vložitev gloVe	**0,9210**	0, 9995	0, 8433
BLSTM	**besedna vložitev word2vec**	**0,8880**	**0,9997**	**0,9248**
C-LSTM	besedna vložitev word2vec	0, 8136	**0,9999**	0, 7811
RCNN	besedna vložitev word2vec	0, 8710	**0,9999**	0, 8648

Tabela 5.10: Klasifikacijska točnost vseh modelov pri klasifikaciji delov člankov na testni podatkovni množici, razdeljeni po vrsti člankov, kjer so r ročno napisani članki, g1 članki, generirani s slabim modelom in g2 članki, generirani z dobrim modelom.

Literatura

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*, 2016.

[2] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[5] fastText. Dosegljivo: `https://fasttext.cc/`. [Dostopano 7. 9. 2018].

[6] fasttext vectors wiki.en.vec. Dosegljivo: `https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.en.vec`. [Dostopano: 14. 8. 2018].

[7] glove.840b.300d.zip. Dosegljivo: `http://nlp.stanford.edu/data/wordvecs/glove.840B.300d.zip`. [Dostopano: 14. 8. 2018].

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[9] Googlenews-vectors-negative300.bin.gz. Dosegljivo: `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/`. [Dostopano: 14. 8. 2018].

[10] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[11] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[12] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[13] Aditi Gupta, Hemank Lamba, and Ponnurangam Kumaraguru. $1.00 per rt #bostonmarathon #prayforboston: Analyzing fake content on twitter. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–12. IEEE, 2013.

[14] M Ikonomakis, S Kotsiantis, and V Tampakas. Text classification: a recent overview. In *Proceedings of the 9th WSEAS International Conference on Computers*, pages 1–6. World Scientific and Engineering Academy and Society (WSEAS), 2005.

[15] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.

[16] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[17] Andrej Karpathy. Convolutional Neural Networks (CNNs / ConvNets) - CS231n Convolutional Neural Networks for Visual Recognition. Dosegljivo: `https://cs231n.github.io/convolutional-networks/`, 2018. [Dostopano 20. 8. 2018].

[18] Andrej Karpathy. Neural Networks - CS231n Convolutional Neural Networks for Visual Recognition. Dosegljivo: `http://cs231n.github.io/neural-networks-1/`, 2018. [Dostopano 14. 12. 2018].

[19] Keras: The Python Deep Learning library. Dosegljivo: `https://keras.io`. [Dostopano 10. 9. 2018].

[20] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[21] Ahmed Khorsi. An overview of content-based spam filtering techniques. *Informatica*, 31(3), 2007.

[22] Philip Koehn. Neural machine translation. *arXiv preprint arXiv:1709.07809*, pages 31–34, 2017.

[23] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273, 2015.

[24] Michael Luca and Georgios Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12):3412–3427, 2016.

[25] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI workshop on Text Learning*, pages 1–7, 1999.

[26] Tomas Mikolov, Quoc Le, V., and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[27] Natural Language Toolkit. Dosegljivo: https://www.nltk.org/. [Dostopano 1. 7. 2018].

[28] Christopher Olah. Conv Nets: A Modular Perspective. Dosegljivo: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/, 2014. [Dostopano 22. 6. 2018].

[29] Christopher Olah. Understanding LSTM Networks. Dosegljivo: http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. [Dostopano 24. 6. 2018].

[30] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[31] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2012.

[32] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[33] Sebastian Raschka. Single-Layer Neural Networks and Gradient Descent. Dosegljivo: https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html, 2015. [Dostopano 15. 8. 2018].

[34] ReLU and Softmax Activation Functions. Dosegljivo: https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions, 2018. [Dostopano 14. 12. 2018].

[35] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[36] Xiaoxin Wei. From recurrent neural network to long short term memory architecture. 2013.

[37] word2vec. Dosegljivo: `https://code.google.com/archive/p/word2vec/`. [Dostopano 7. 9. 2018].

[38] Ziang Xie. Neural text generation: A practical guide. *arXiv preprint arXiv:1711.09534*, 2017.

[39] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[40] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.

[41] Xiaojin Zhu. CS769 Spring 2010 Advanced Natural Language Processing: Basic Text Process. Dosegljivo: `http://pages.cs.wisc.edu/~jerryzhu/cs769/text_preprocessing.pdf`, 2010. [Dostopano 26. 6. 2018].

[42] Jaka Čibej, Špela Holdt, Arhar, Tomaž Erjavec, and Katja Zupan. Smernice za označevanje računalniško posredovane komunikacije: tokenizacija, stavčna segmentacija, normalizacija, lematizacija in oblikoskladenjsko označevanje. Dosegljivo: `http://nl.ijs.si/janes/wp-content/uploads/2014/09/Janes-smernice-v1.0.pdf`, 2016. [Dostopano 26. 6. 2018].